

# Addition Operators Solutions

- What is the prototype of the addition operator?

`T operator +(const T& lhs, const T& rhs);` // Addition operator for type T

- How is it invoked?

- Whenever we write a statement such as

`a + b;`

- The compiler will generate code which calls the operator with the appropriate argument
- The operator is a non-member function, so it will be called as  
`operator+(a, b);`

- What is the prototype of the += operator?

T operator +=(const T& rhs); // Compound addition operator for type T

- Whenever we write a statement such as

a += b;

- The compiler will generate code which calls the operator with the appropriate argument
- The operator is a member function, so it will be called as

a.operator+=(b);

- Write an implementation of the plus operator that does not make use of the += operator

```
T operator + (const T& lhs, const T& rhs) {  
    T temp {lhs};           // Make a copy of the lhs argument  
    temp += rhs;            // Add the rhs argument to it  
    return temp;            // Return the result  
}
```

- What are the advantages of expressing the plus operator this way?
  - Code reuse

- Explain what this code does

```
return *this;
```

- The += operator should return the modified value of the object
- Because the += operator is a member function, it is passed the “this” pointer as an argument
- “this” is a pointer to the object the function was called on
- By dereferencing this, we get the current value of the object, after we have modified it
- Therefore \*this returns the modified value of the object